CANopen BASICS

TRINAMIC

Now part of Analog Devices

CANOPEN BASICS

AGENDA

1. CAN Bus
2. CANopen
    a) Basic
    b) Communication
    c) Network Management
    d) CIA - 402

- CAN BUS

Now part of Analog Devices

CAN BUS

AGENDA

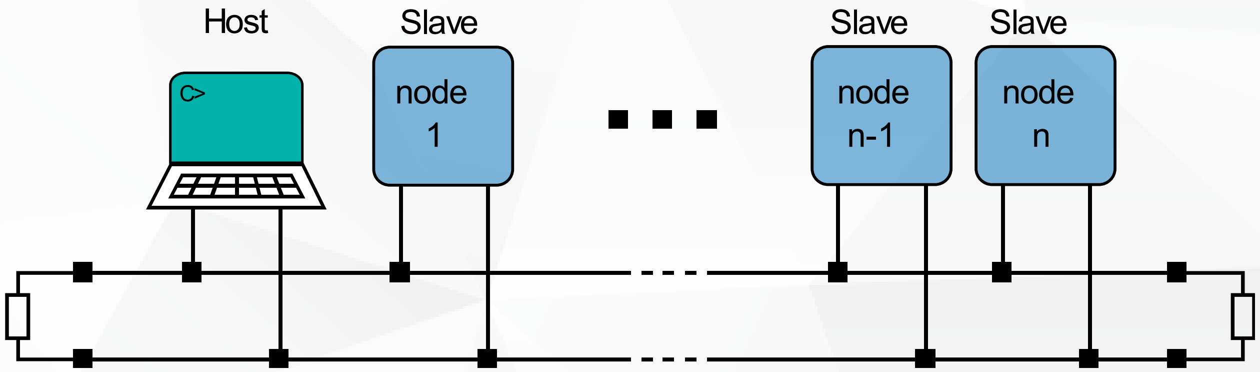
1. Basics
2. Datagram
3. Pitfalls

CAN BUS

BASICS

- CAN: Controller Area Network
- Highly flexible multi-master protocol
- Broadcasts of frames to max. 127 nodes
- Hardware-level arbitration by ID avoids collisions

- Data rates up to 1MBit/s
  - Typical: 20kBit/s, 50kBit/s, 100kBit/s, 125kBit/s, 250kBit/s, 500kBit/s, 1MBit/s
- Typically Sub-D 9
  - 2=CAN_LOW, 7=CAN_HIGH, 3=GND, 6=optional GND
- CAN bus requires special adapters to PC/MAC
  - usually with proprietary API, no standard

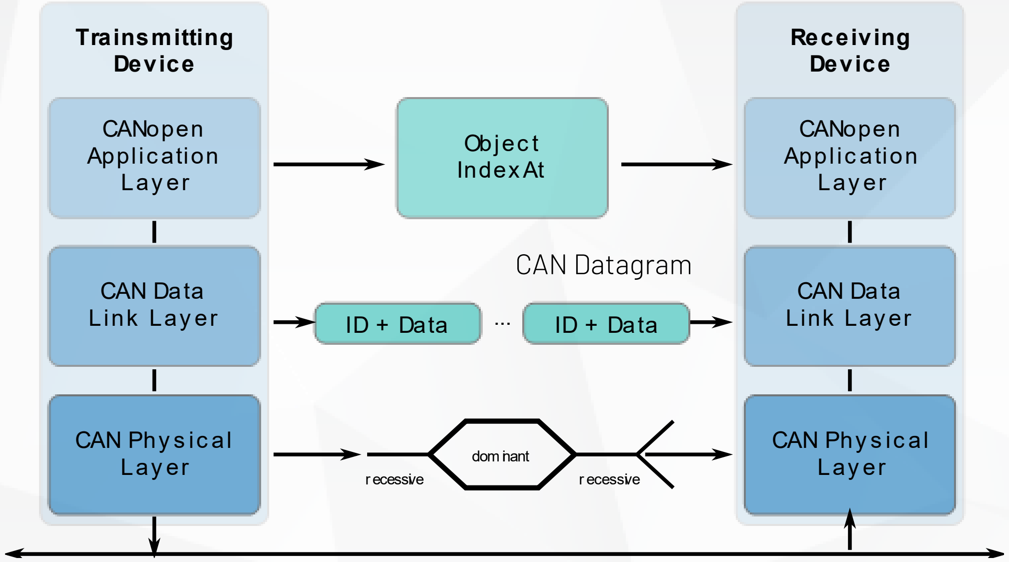


CAN BUS

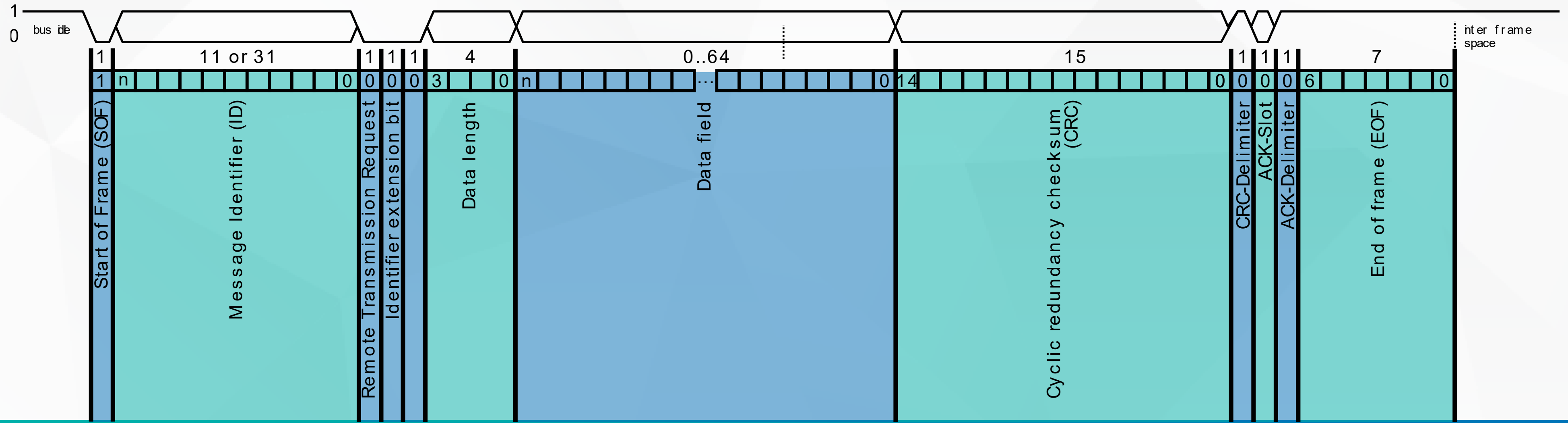
DATAGRAM

- Arbitration IDs 11 or 29 bits + control bits
  - Bus with lowest ID may send prioritized
- Max 8 bytes of data (max 64 bytes with CAN-FD)
  - Transport of application layer data (e.g. CANopen or TMCL)
- Build-in CRC
- Build-in Acknowledge and retry on unacknowledged datagrams

Protocol Layer Instructions, e.g. CANopen



CAN Datagram

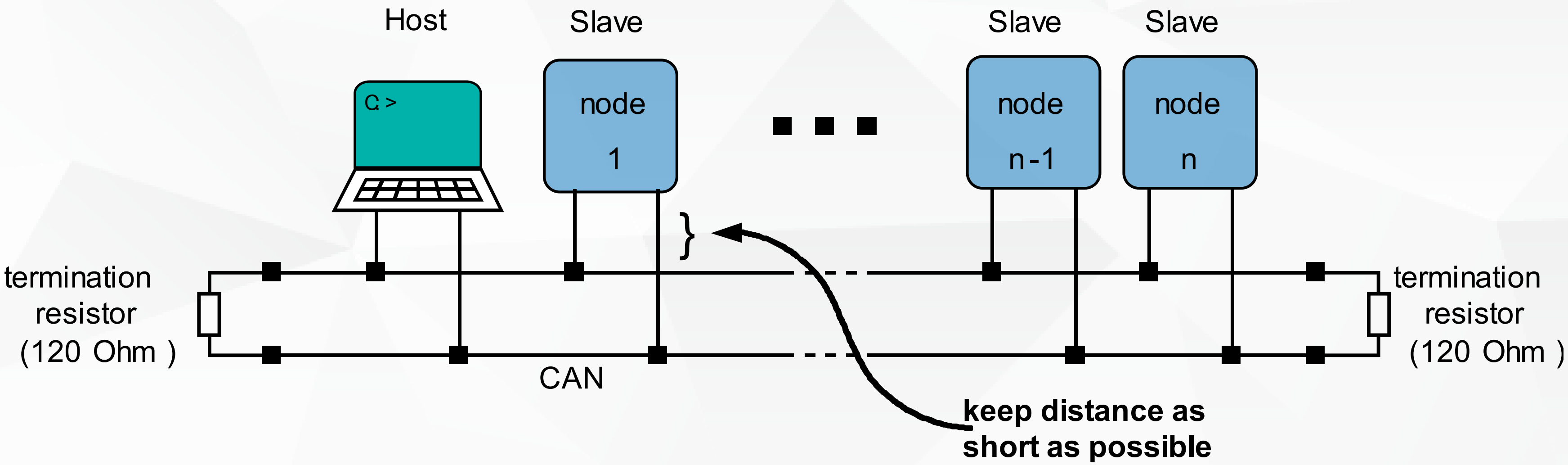


CAN BUS

PITFALLS

- Missing 120R termination
- Non-linear/bus busses, e.g., star topology with many nodes
- Overall length of the bus determines maximum usable bitrate
- Wrong CAN adapter for toolchain
  - As there is no standard for CAN adapters, software tools like the TMCL-IDE can only support a limited number of adapters

Bitrate	Cable Length
10 kbits/s	6.7 km
20 kbits/s	3.3 km
50 kbits/s	1.3 km
125 kbits/s	530 m
250 kbits/s	270 m
500 kbits/s	130 m
1 Mbits/s	40 m



- CANopen

TRINAMIC

Now part of Analog Devices

CANopen - BASICS

AGENDA

1. Basics
2. Relevant Standards for Trinamic Products

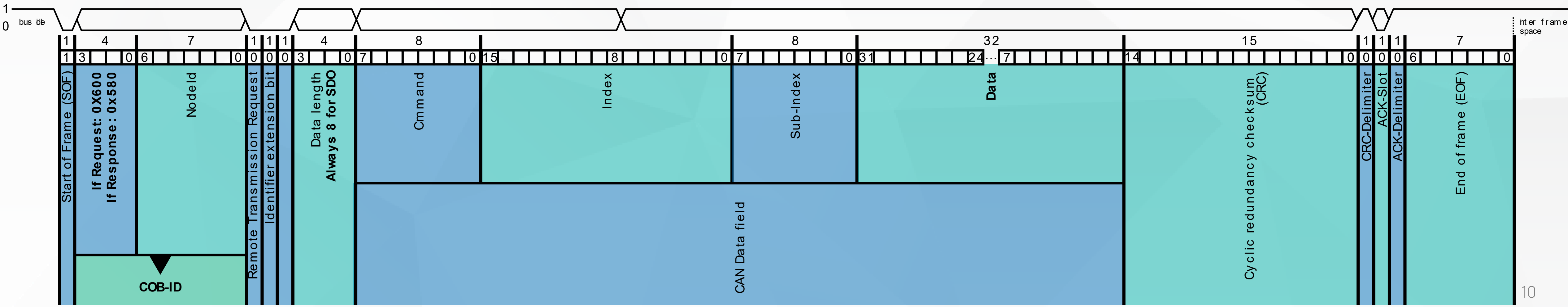
CANopen

BASICS

- Goal: Standardized access to generic devices (nodes) with application specific function profiles
- Standardized by **C**AN **i**n **A**utomation consortium (CiA)
- DS301 standard since 1995

- DS301 defines CANopen protocol, which is utilized by application specific protocols
- CAN ID (here **COB-ID**) represents the node id and a function code
- Strict master-slave principle (exception: flying master)

CANopen Datagram



CANopen

RELEVANT STANDARDS FOR TRINAMIC PRODUCTS

Standard	Context	Comment
CiA 301	Basic CANopen protocol	
CiA 303	Control of on-board LEDs	LEDs are standardized
CiA 307	Electronical Data Sheet (EDS)	XML file describing objects for device, comes with firmware
CiA 402	Drives and Motion Control Profile	Minimum function set for stepper and servo drives
Might come across:		
CiA 304	Safety Relevant Communication	Alias EN50325-5
DSP 454	Profile for Energy Management Systems	e.g. Pedelecs

Conformance tests can prove the compatibility with standards – usually just CiA 301

CANOPEN - COMMUNICATION

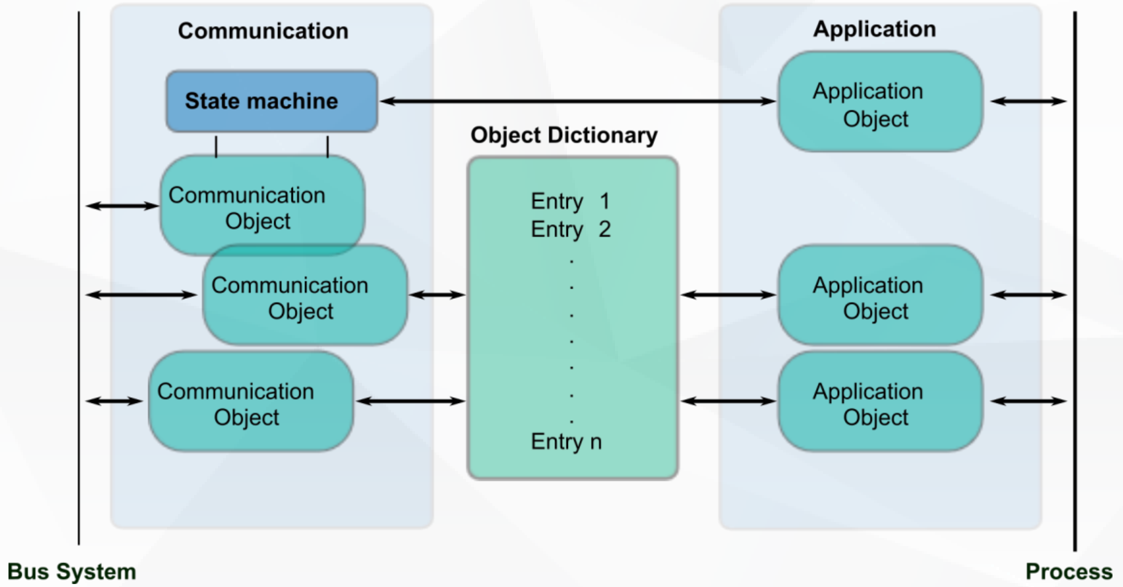
AGENDA

1. Objects
2. Area/Object Examples
3. Service Data Objects
4. SDO: Transfer Types
5. PDO: Process Data Objects
6. PDO Settings
7. SDO vs. PDO
8. Example

CANopen - COMMUNICATION

OBJECTS

- All parameters and functions mapped to objects
- All objects are mapped to an **Object Dictionary** structuring the data (memory map) with index and sub-index
  - Index: 16Bit
  - Sub-Index: 8Bit
- Objects can have different data types
  - e.g. INT32, STRING, BYTE…
- **SDO** and **PDO** communication mechanisms allow accessing objects
  - **SDO**: **S**ervice **D**ata **O**bjects
  - **PDO**: **P**rocess **D**ata **O**bjects
- Specific areas within the **Object Dictionary** are dedicated to certain purpose



CANopen - COMMUNICATION

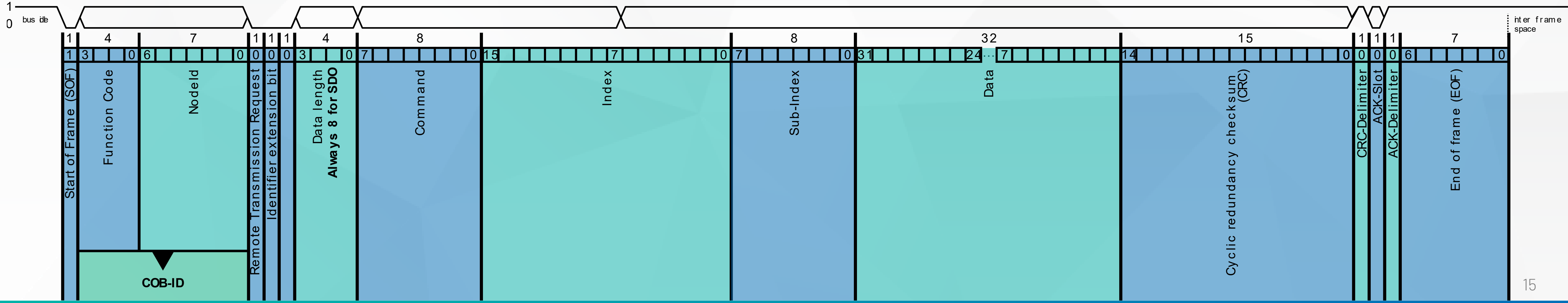
AREA/OBJECT EXAMPLES

	Indexes	Context	Comment
Areas	0x0000	Not used	
	0x0001...0x001F	Static dara types	
	0x0020...0x003F	Complex data types.	
	0x0040...0x005F	Manurfacturer specific complex data types	
	0x0060...0x007F	Device profile specific complex data types	
	0x00A0...0x0FFF	Reserved for further use	
	0x1000...0x1FFF	Communication profile area	Communication specific parameters, according to CiA 301
	0x2000...0x5FFF	Manufacturer specific profile area	E.g. non-standard Trinamic parameters/functions/features
	0x6000...0x9FFF	Standardized device profile area	E.g. CiA402 Motion Control Profile specific objects
	0xA000...0xBFFF	Standardized device communication area	E.g. PDO configuration
	0xC000...0xFFFF	Reserved for further use	
Objects	0x6040	Control Word UINT16	Controls CiA402 state machine
	0x6060	Modes of Operation INT8	E.g. 1=Profile Position (pp), 3=Profile Velocity (pv), 6=Homing (hm), 10=Cyclic Synchronous Torque (cst))

CANopen - COMMUNICATION

SDO: SERVICE DATA OBJECTS

- All objects can be accessed via SDOs
- SDO accesses are always acknowledged
- COB-ID defines access type and node ID
    - Access Types: Read (=download) or Write (=upload)

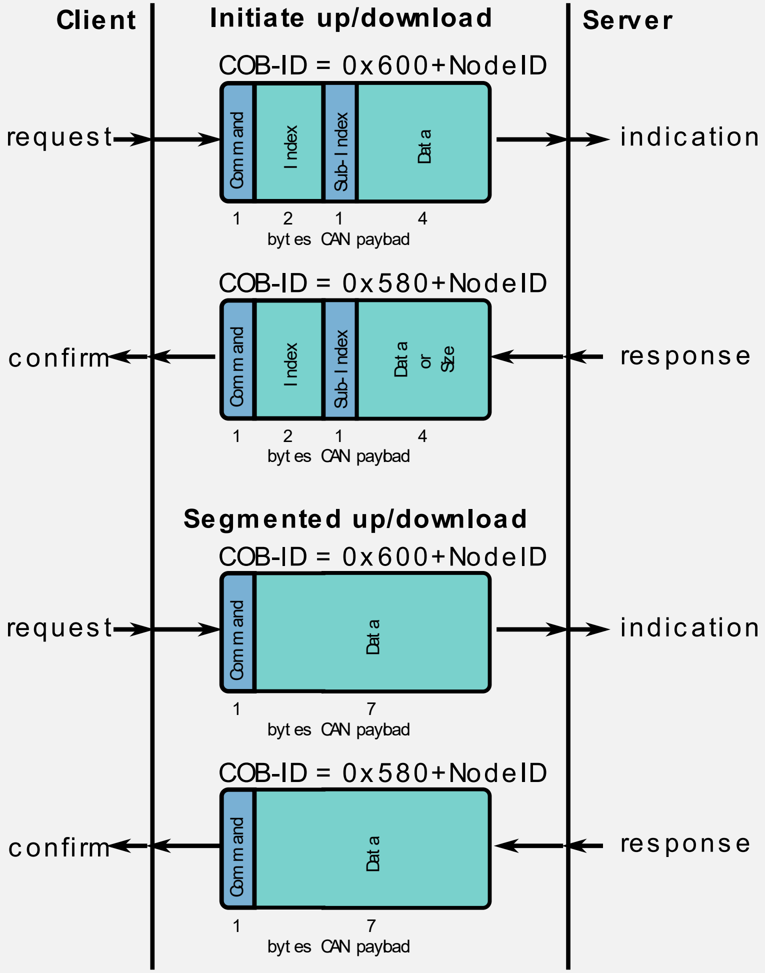


CANopen - COMMUNICATION

SDO: TRANSFER TYPES

- Expedited Transfer: 4-byte object transfer
  - One 8-byte CANopen frame for sending
  - One 8-byte CANopen frame as acknowledge

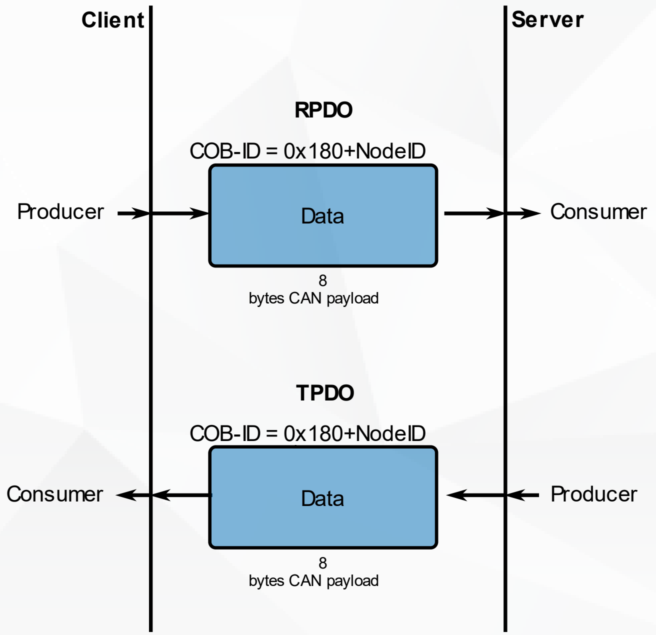
- Segmented Transfer: n>4-byte-transfer
  - Segmentation to 8-byte-frames
  - Each frame is acknowledged



CANopen - COMMUNICATION

PDO: PROCESS DATA OBJECTS

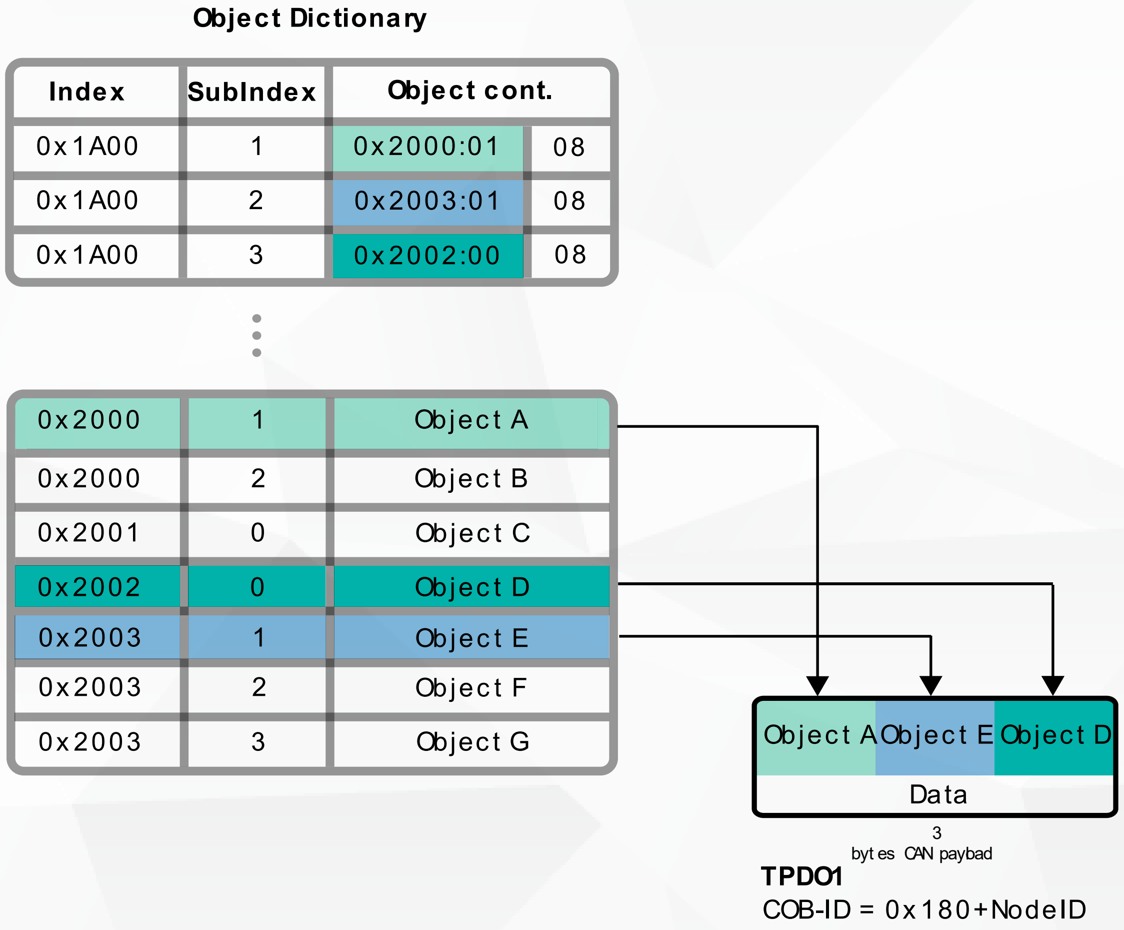
- Fastest data-exchange possible
  - single CAN frame with 1...8 bytes
  - no communication overhead
  - no ACK
- RPDOs allow writing objects to a device, either:
  - Asynchronous
  - Synchronised
- TPDOs are transmitted automatically by a device, either:
  - Frequently (Time triggered)
  - Asynchronous
  - Synchronised
- Specific objects can be mapped to PDOs



CANopen - COMMUNICATION

PDO SETTINGS

- Settings for PDOs in Communication Profile Area:
  - RPDO communication settings: 0x1400…0x15FF
  - RPDO mapping: 0x1600…0x17FF
  - TPDO communication settings: 0x1800…0x19FF
  - TPDO mapping: 0x1A00…0x1BFF
- Proper definition of PDOs allows inter-device-communication
  - TPDO of one device can be the RPDO of another device
  - E.g. analog input measured on one device can define the target velocity for a second device



Object Dictionary

Index	SubIndex	Object cont.	
0x1A00	1	0x2000:01	08
0x1A00	2	0x2003:01	08
0x1A00	3	0x2002:00	08

0x2000	1	Object A
0x2000	2	Object B
0x2001	0	Object C
0x2002	0	Object D
0x2003	1	Object E
0x2003	2	Object F
0x2003	3	Object G

TPDO1
COB-ID = 0x180+NodeID

CANopen - COMMUNICATION

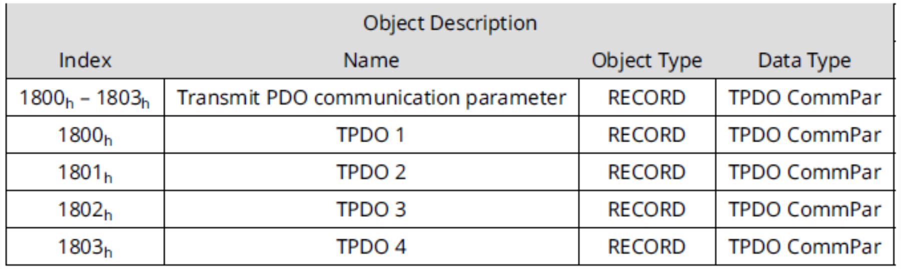
PDO SETTINGS

- Settings for PDOs in **Communication Profile Area**:
  - **TPDO communication settings: 0x1800…0x19FF**
  - TPDO mapping: 0x1A00…0x1BFF

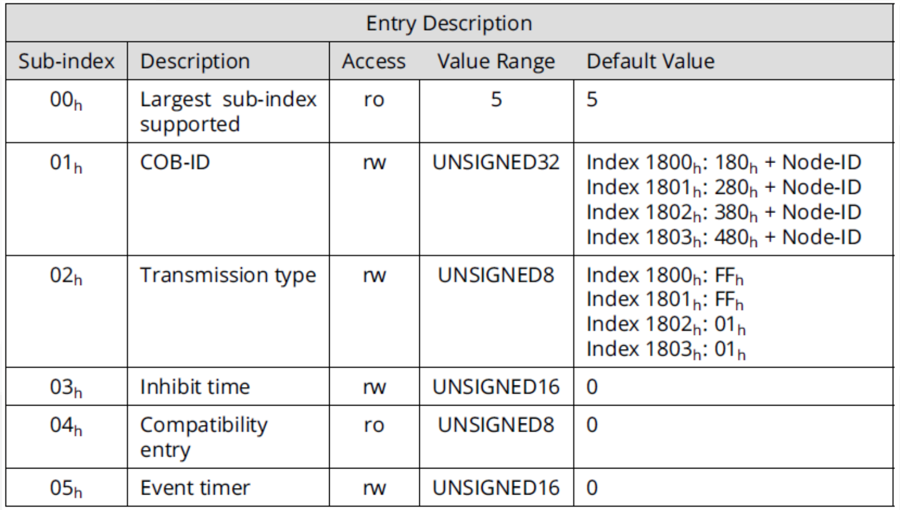
- Usually there is a predefined set of COB-IDs which can be changed:
  - **TPDO**: 0x180+NodeID, 0x280+NodeID , …, 0x480+NodeID
  - **RPDO**: 0x200+NodeID, 0x300+NodeID, …, 0x500+NodeID

NodeID = 1-127

EXAMPLE – TPDO COMMUNICATION SETTINGS



Object Description			
Index	Name	Object Type	Data Type
1800$_h$ – 1803$_h$	Transmit PDO communication parameter	RECORD	TPDO CommPar
1800$_h$	TPDO 1	RECORD	TPDO CommPar
1801$_h$	TPDO 2	RECORD	TPDO CommPar
1802$_h$	TPDO 3	RECORD	TPDO CommPar
1803$_h$	TPDO 4	RECORD	TPDO CommPar



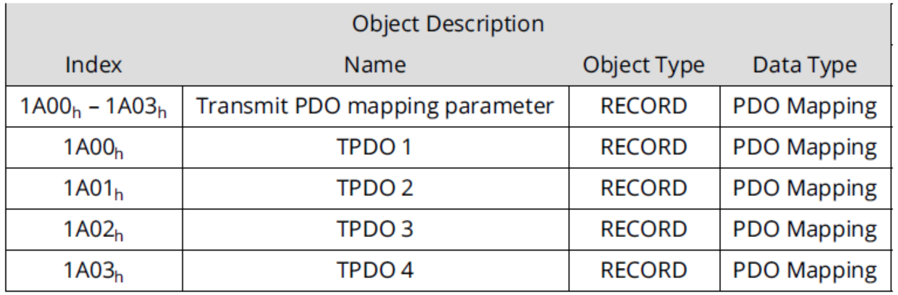
Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00$_h$	Largest sub-index supported	ro	5	5
01$_h$	COB-ID	rw	UNSIGNED32	Index 1800$_h$: 180$_h$ + Node-ID Index 1801$_h$: 280$_h$ + Node-ID Index 1802$_h$: 380$_h$ + Node-ID Index 1803$_h$: 480$_h$ + Node-ID
02$_h$	Transmission type	rw	UNSIGNED8	Index 1800$_h$: FF$_h$ Index 1801$_h$: FF$_h$ Index 1802$_h$: 01$_h$ Index 1803$_h$: 01$_h$
03$_h$	Inhibit time	rw	UNSIGNED16	0
04$_h$	Compatibility entry	ro	UNSIGNED8	0
05$_h$	Event timer	rw	UNSIGNED16	0

CANOPEN – COMMUNICATION

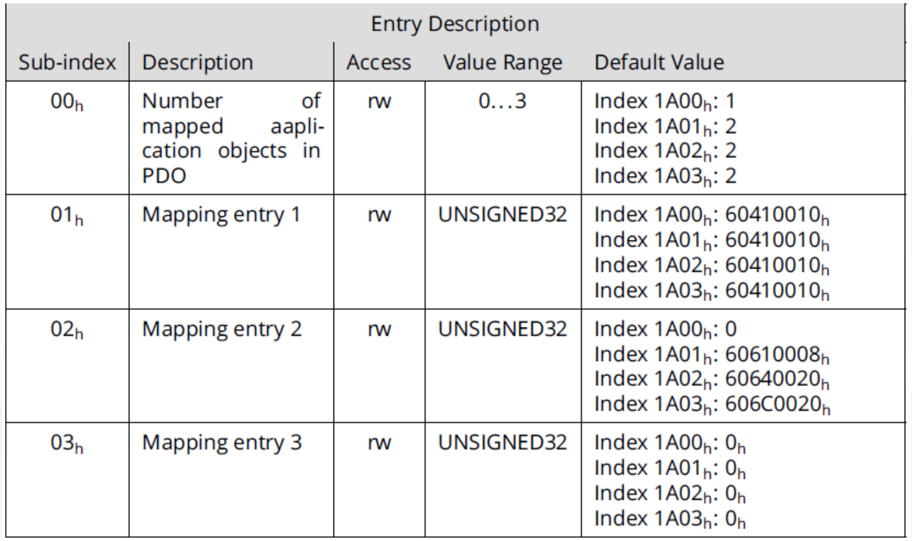
PDO SETTINGS

- Settings for PDOs in **Communication Profile Area**:
  - TPDO communication settings: 0x1800…0x19FF
  - TPDO mapping: 0x1A00…0x1BFF

EXAMPLE – TPDO MAPPING



Object Description			
Index	Name	Object Type	Data Type
1A00$_h$ – 1A03$_h$	Transmit PDO mapping parameter	RECORD	PDO Mapping
1A00$_h$	TPDO 1	RECORD	PDO Mapping
1A01$_h$	TPDO 2	RECORD	PDO Mapping
1A02$_h$	TPDO 3	RECORD	PDO Mapping
1A03$_h$	TPDO 4	RECORD	PDO Mapping



Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00$_h$	Number of mapped application objects in PDO	rw	0…3	Index 1A00$_h$: 1 Index 1A01$_h$: 2 Index 1A02$_h$: 2 Index 1A03$_h$: 2
01$_h$	Mapping entry 1	rw	UNSIGNED32	Index 1A00$_h$: 60410010$_h$ Index 1A01$_h$: 60410010$_h$ Index 1A02$_h$: 60410010$_h$ Index 1A03$_h$: 60410010$_h$
02$_h$	Mapping entry 2	rw	UNSIGNED32	Index 1A00$_h$: 0 Index 1A01$_h$: 60610008$_h$ Index 1A02$_h$: 60640020$_h$ Index 1A03$_h$: 606C0020$_h$
03$_h$	Mapping entry 3	rw	UNSIGNED32	Index 1A00$_h$: 0$_h$ Index 1A01$_h$: 0$_h$ Index 1A02$_h$: 0$_h$ Index 1A03$_h$: 0$_h$

CANOPEN - COMMUNICATION

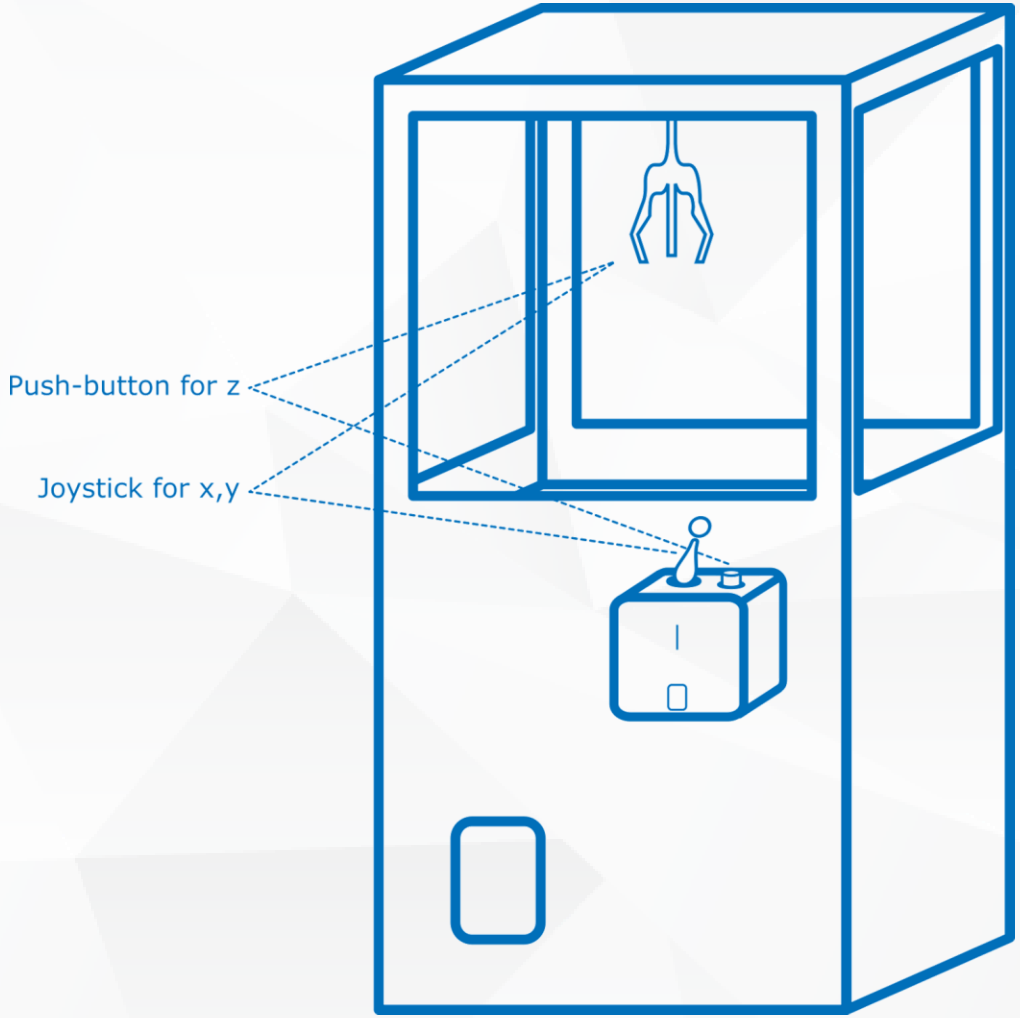
SDO vs. PDO

SDO	PDO
As per standard, all CANopen nodes must implement a server for write/read access via **SDOs** as communication interface	As per standard, all CANopen nodes must be able to auto-transmit process data via **PDOs**.
Access to all objects	Access to specific mappable objects
Master (Client) accesses a single node (Server) for reading/writing and receives answer from a single device.	Auto read/write PDO data to master or other nodes, no acknowledge
Quite high communication overhead	**No communication overhead**
Typically for configuration	**Typically for application/process data**

CANopen - COMMUNICATION

EXAMPLE

- A claw machine implements different nodes for music, payment and movement.
- The master takes control of all nodes and parametrizes them via SDO.
- The joystick sends a speed value each 100ms via **timer TPDO**. PANdrives for x and y movement of the gripper consume the data and use it as position input.
- If the push-button is hit, a **triggered TPDO** is produced and consumed by the PANdrive to start the z movement.
- As soon as the z-axis PANdrive reached a certain position a **triggered TPDO** is produced, consumed by another PANdrive for the gripper, which starts to close the claw.



CANopen - NETWORK MANAGEMENT

AGENDA

1. NMT Network Management
2. EMCY: Emergency Messages
3. Surveillance

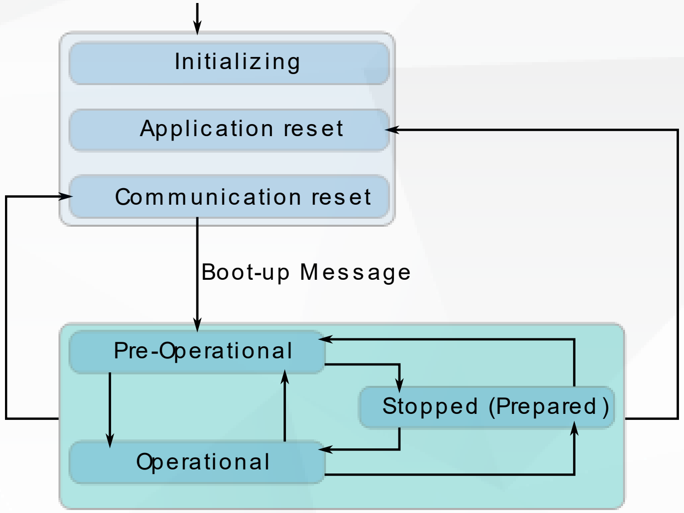
CANopen - NETWORK MANAGEMENT

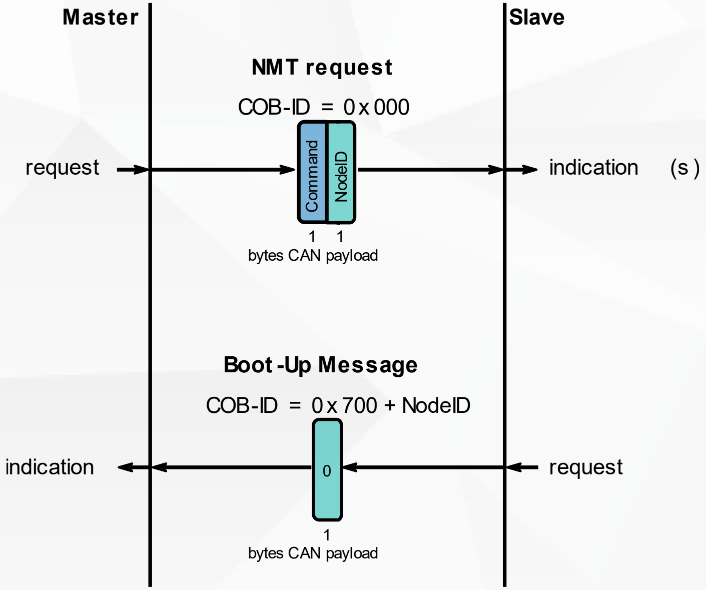
NMT: NETWORK MANAGEMENT

- **NMT** frame with COB-ID = 0 (highest priority) allows changing states
- Each device implements three states:

	Allowed Operations				
State	NMT	EMCY	SDO	PDO	Green CANopen run LED
STOPPED	x				Single flash; 200ms on, 1000ms off
PRE-OP	x	x	x		Blinking; 200ms on/off
OP	x	x	x	x	On

- Nodes start **PRE-OP** after (power-on) reset and confirm readiness with boot-up message

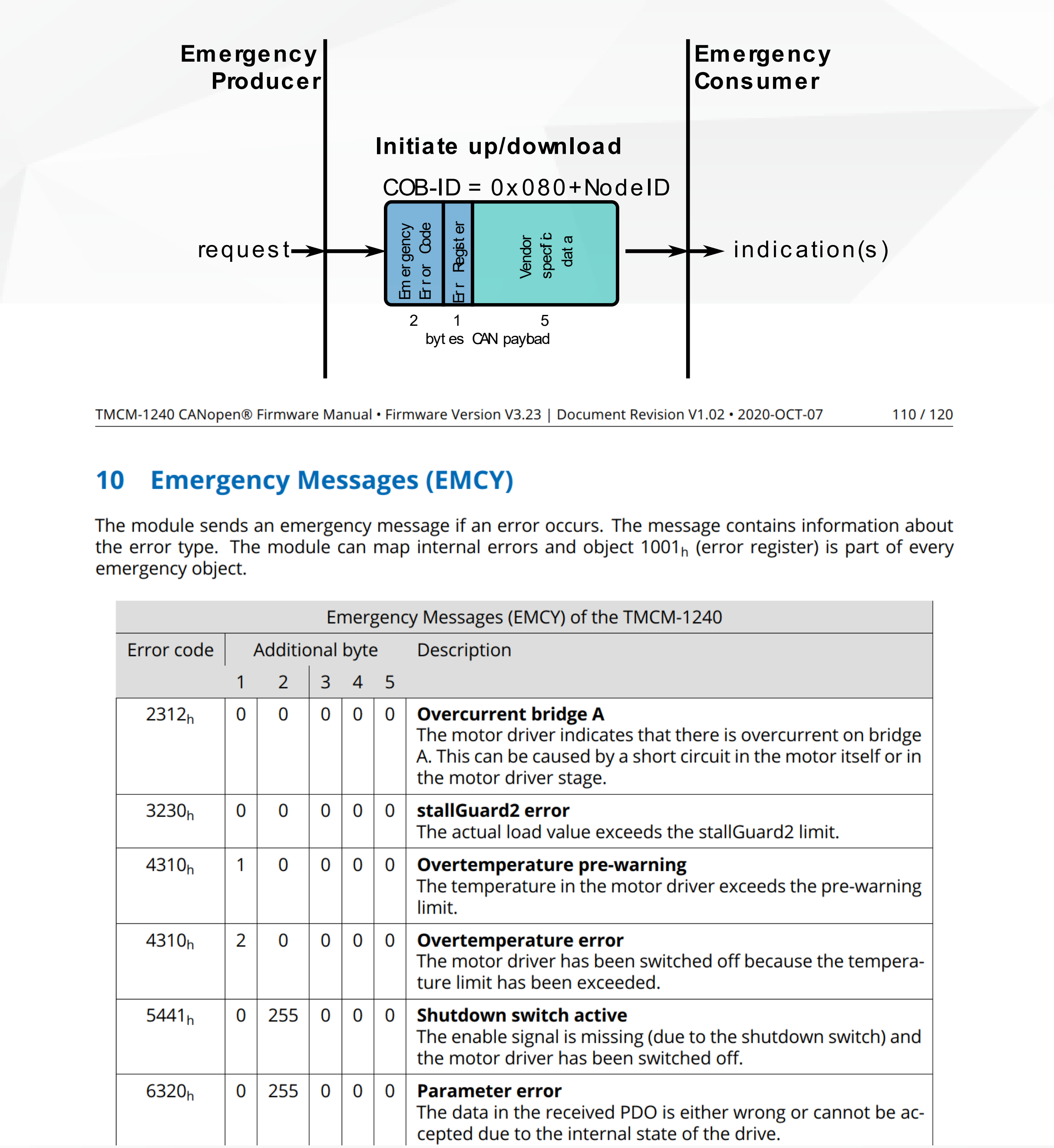




CANopen – EMERGENCY MESSAGES

EMCY: EMERGENCY MESSAGES

- EMCY messages allow indicating errors
  - COB-ID = free to define (object 0x1014, usually 0x80+NodeID, thus it's high priority)
  - eight bytes indicating an error (e.g. ref switch hit)
- The handling of errors can be specified
  - e.g. objekt 0x1029: Error Behaviour



TMCM-1240 CANopen® Firmware Manual • Firmware Version V3.23 | Document Revision V1.02 • 2020-OCT-07 110 / 120

10 Emergency Messages (EMCY)

The module sends an emergency message if an error occurs. The message contains information about the error type. The module can map internal errors and object 1001$_h$ (error register) is part of every emergency object.

Emergency Messages (EMCY) of the TMCM-1240

Error code	Additional byte					Description
	1	2	3	4	5	
2312$_h$	0	0	0	0	0	**Overcurrent bridge A** The motor driver indicates that there is overcurrent on bridge A. This can be caused by a short circuit in the motor itself or in the motor driver stage.
3230$_h$	0	0	0	0	0	**stallGuard2 error** The actual load value exceeds the stallGuard2 limit.
4310$_h$	1	0	0	0	0	**Overtemperature pre-warning** The temperature in the motor driver exceeds the pre-warning limit.
4310$_h$	2	0	0	0	0	**Overtemperature error** The motor driver has been switched off because the temperature limit has been exceeded.
5441$_h$	0	255	0	0	0	**Shutdown switch active** The enable signal is missing (due to the shutdown switch) and the motor driver has been switched off.
6320$_h$	0	255	0	0	0	**Parameter error** The data in the received PDO is either wrong or cannot be accepted due to the internal state of the drive.

CANopen - SURVEILLANCE

SURVEILLANCE

- Trinamic devices support **Heart Beat Messages** to allow master and devices to monitor another and identify potential defects
- Heart Beat Producer
  - Generates **Heart Beat Messages** in configurable frequency
- Heart Beat Consumer
  - Expecting and monitoring **Heart Beat Messages** from other devices
- Heart Beat Messages contain the NMT state

CANopen - CIA-402

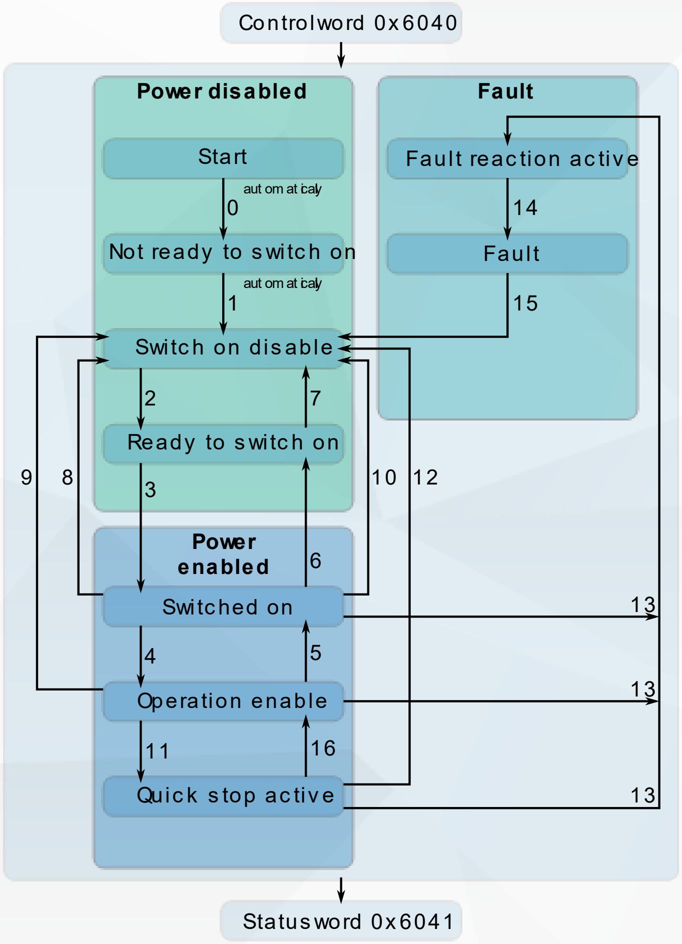
1. Drives and Motion Control Profile
2. Important Objects
3. Important Manufacturer Specific Objects
4. Start Motor

CANopen - CIA-402

DRIVES AND MOTION CONTROL PROFILE

- Specifies objects for minimum function set and modes of operation Servo Drives
  - Applicable for BLDC, Steppers
  - More specific functions accessible via **Manufacturer Specific** objects → reduces replaceability of devices
- Each axis is controlled by a dedicated state machine
  - Controlword (0x6040) and Statusword (0x6041)
  - State machine not related to NMT, exception: NMT→STOPPED results in CIA-402 FAULT state

CIA-402 STATE MACHINE



CANopen - CIA-402

IMPORTANT OBJECTS

Index	Name	Description
0x6040	Control Word	Controls CiA-402 state machine
0x6060	Modes of Operation	1 = Profile Position (pp) 3 = Profile Velocity (pv) 6 = Homing (hm) 10 = Cyclic Synchronous Torque (cst)
0x607A	Target Position	Target position in mode pp
0x60ff	Target Velocity	Target velocity in mode pv
0x6098	Homing Method	Reference search method in mode hm
0x6071	Target Torque	Target torque in mode cst
0x6064	Position Actual Value	Actual position
0x606C	Velocity Actual Value	Actual velocity
0x6041	Status Word	Actual state of CiA state machine

CANopen - CIA-402

IMPORTANT MANUFACTURER SPECIFIC OBJECTS (TRINAMIC)

Index	Name	Description
0x2000	Microstep resolution	Microstep resolution
0x2003	Absolute max. current	Run current
0x2004	Standby current	Current when the object ist standing
0x2005	Switch Parameters	Configuration of stop switches
Plus others for StallGuard, CoolStep…		

CANopen - START A MOTOR

	Object	Name	Value	Description
Prepare Operation	0x2005	Limit Switches	3	Deactivate limit switches
	0x6040	Controlword	6	State transition to Shut Down
	0x6040	Controlword	7	State transition to Switched On
	0x6040	Controlword	15	State transition to Enable Operation
Position Mode	0x6060	Modes of operation	1	To Profile Position Mode (pp)
	0x607A	Target Position	<target position>	Set Target Position
	0x6040	Controlword	31	Set state machine to Enable Operation, Reset Faults, Motor starts running to target position and then stops.
	0x6040	Controlword	15	Set state machine to Enable Operation
Velocity Mode	0x6060	Modes of operation	3	To Velocity Mode (vl)
	0x60FF	Target Velocity	<target velocity>	Set target velocity